

**Gates & Cooper** *LLP*

Howard Hughes Center  
6701 Center Drive West, Suite 1050  
Los Angeles, California 90045

**RECEIVED**  
CENTRAL FAX CENTER

**AUG 01 2005**

**FAX TRANSMISSION TO USPTO**

TO: Commissioner for Patents  
Attn: Examiner Mary J. Steelman  
Patent Examining Corps  
Facsimile Center  
Alexandria, VA 22313-1450

FROM: George H. Gates  
OUR REF.: STL9-2000-0084US1  
TELEPHONE: (310) 642-4146

Total pages, including cover letter: 18

PTO FAX NUMBER: (571) 273-8300

If you do NOT receive all of the pages, please telephone us at (310) 641-8797, or fax us at (310) 641-8798.

Title of Document Transmitted:	BRIEF OF APPELLANTS AND AUTHORIZATION TO CHARGE DEPOSIT ACCOUNT NO. 09-0460 OF IBM IN THE AMOUNT OF \$500.00 FOR FILING FEE.
Applicant:	Rebecca Lau Poole et al.
Serial No.:	09/939,813
Filed:	August 27, 2001
Group Art Unit:	2122
Title:	TOPOLOGICAL MULTI-TIER BUSINESS APPLICATION COMPOSER
Our Ref. No.:	STL9-2000-0084US1

Please charge all fees to Deposit Account No. 09-0460 of IBM Corporation, the assignee of the present application.

By: 

Name: George H. Gates  
Reg. No.: 33,500

I hereby certify that this paper is being transmitted by facsimile to the U.S. Patent and Trademark Office on the date shown below.

  
Signature

8/1/05  
Date

Confirmation No.: 2901  
Due Date: August 1, 2005IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants:	Rebecca Lau Poole et al.	Examiner:	Mary J. Steelman
Serial No.:	09/939,813	Group Art Unit:	2122
Filed:	August 27, 2001	Docket:	STL9-2000-0084US1
Title:	TOPOLOGICAL MULTI-TIER BUSINESS APPLICATION COMPOSER		

**CERTIFICATE OF MAILING OR TRANSMISSION UNDER 37 CFR 1.8**I hereby certify that this correspondence is being filed via facsimile transmission to the U.S. Patent and Trademark Office on August 1, 2005.By: 

Name: George H. Gates

**MAIL STOP APPEAL BRIEF - PATENTS**

Commissioner for Patents

P.O. Box 1450

Alexandria, VA 22313-1450

Dear Sir:

We are transmitting herewith the attached:

- ☒ Transmittal sheet, in duplicate, containing a Certificate of Mailing or Transmission under 37 CFR 1.8.
- ☒ Brief of Appellant(s).
- ☒ Charge the Fee for the Brief of Appellant(s) in the amount of \$500.00 to the Deposit Account.

Please consider this a **PETITION FOR EXTENSION OF TIME** for a sufficient number of months to enter these papers, if appropriate.

Please charge all fees to Deposit Account No. 09-0460 of IBM Corporation, the assignee of the present application. A duplicate of this paper is enclosed.

Customer Number 22462**GATES & COOPER LLP**

Howard Hughes Center

6701 Center Drive West, Suite 1050

Los Angeles, CA 90045

(310) 641-8797

By: 

Name: George H. Gates

Reg. No.: 33,500

GHG/sjm

**RECEIVED**  
**CENTRAL FAX CENTER**

AUG 01 2005

Due Date: August 1, 2005

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**  
**BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re Application of:

Inventor: Rebecca Lau Poole et al.

Serial # : 09/939,813

Filed: August 27, 2001

Title: TOPOLOGICAL MULTI-TIER BUSINESS  
APPLICATION COMPOSER

Examiner: Mary J. Steelman

Group Art Unit: 2122

Appeal No.: \_\_\_\_\_

**BRIEF OF APPELLANT'S****MAIL STOP APPEAL BRIEF - PATENTS**

Commissioner for Patents

P.O. Box 1450

Alexandria, VA 22313-1450

Dear Sir:

In accordance with 37 CFR §41.37, Appellants' attorney hereby submits the Brief of Appellants on appeal from the final rejection in the above-identified application, as set forth in the Office Action dated March 1, 2005.

Please charge the amount of \$500 to cover the required fee for filing this Appeal Brief as set forth under 37 CFR §41.37(a)(2) and 37 CFR §41.20(b)(2) to Deposit Account No. 09-0460 of IBM Corporation, the assignee of the present application.

Also, please charge any additional fees or credit any overpayments to Deposit Account No. 09-0460 of IBM Corporation.

08/02/2005 TL0111 00000035 090460 09939813  
01 FC:1402 500.00 DA**I. REAL PARTY IN INTEREST**

The real party in interest is International Business Machines Corporation, the assignee of the present application.

## II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences for the above-referenced patent application.

## III. STATUS OF CLAIMS

Claims 1-24 are pending in the application.

Claims 1-24 were rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,208,345 to Sheard et al. (Sheard), in view of U.S. Patent No. 6,854,107 to Green et al. (Green).

Claims 1-24 are being appealed.

## IV. STATUS OF AMENDMENTS

Amendments to claims 1, 9 and 17 under 37 C.F.R. §1.116 were submitted subsequent to the final Office Action on April 29, 2005. An Advisory Action mailed June 6, 2005 indicated that these amendments would be entered for the purposes of appeal.

## V. SUMMARY OF CLAIMED SUBJECT MATTER

Briefly, Appellants' invention, as recited in independent claims 1, 9, and 17, are generally directed to developing multi-tier business applications.

Independent claim 1 recites a computer-implemented system. The system comprises an Integrated Development Environment (IDE), executed by a computer, for creating and maintaining a multi-tier business application on a multiple tier computer network, wherein the IDE includes a Topological Multi-Tier Business Application Composer that is used by a developer to graphically create and maintain the multi-tier business application, the Composer includes a window and a palette, the palette contains graphical constructs representing tiers and components of the tiers that are used to create and maintain a graphical representation of the multi-tier business application in the window, and when creating the multi-tier business application, the developer decides on a number of tiers, identifies workstations and servers within each of the tiers, and defines processing performed by each tier and its components.

Independent claim 9 recites a computer-implemented method for developing multi-tier business applications. The method comprises creating and maintaining a multi-tier business application on a multiple tier computer network using an Integrated Development Environment

(IDE) executed by a computer, wherein the IDE includes a Topological Multi-Tier Business Application Composer that is used by a developer to graphically create and maintain the multi-tier business application, the Composer includes a window and a palette, the palette contains graphical constructs representing tiers and components of the tiers that are used to create and maintain a graphical representation of the multi-tier business application in the window, and when creating the multi-tier business application, the developer decides on a number of tiers, identifies workstations and servers within each of the tiers, and defines processing performed by each tier and its components.

Independent claim 17 recites an article of manufacture embodying logic for developing multi-tier business applications. The logic comprises creating and maintaining a multi-tier business application on a multiple tier computer network using an Integrated Development Environment (IDE) executed by a computer, wherein the IDE includes a Topological Multi-Tier Business Application Composer that is used by a developer to graphically create and maintain the multi-tier business application, the Composer includes a window and a palette, the palette contains graphical constructs representing tiers and components of the tiers that are used to create and maintain a graphical representation of the multi-tier business application in the window, and when creating the multi-tier business application, the developer decides on a number of tiers, identifies workstations and servers within each of the tiers, and defines processing performed by each tier and its components.

With regard to the claims, Appellants' attorney requests that the Board refer to the specification generally. Specific portions of the specification that directly relate to the claims on appeal include:

- (a) at page 3, line 23 through page 4, line 14;
- (b) at page 5, line 13 through page 6, line 7, and in FIG. 2 as reference numbers 28-38;
- (c) at page 6, line 8 through page 7, line 29, and in FIG. 3 as reference numbers 40-44; and
- (d) at page 10, line 22 through page 12, line 6, and in FIG. 6 as reference numbers 60-72.

## VI. GROUND OF REJECTION TO BE REVIEWED ON APPEAL

1. Whether claims 1-24 are unpatentable under 35 U.S.C. § 103(a) as being rendered unpatentable over U.S. Patent No. 6,208,345 to Sheard, in view of U.S. Patent No. 6,854,107 to

Green.

## VII. ARGUMENTS

### A. The Office Action Rejections

In paragraph (4) of the Office Action, claims 1-24 were rejected under 35 U.S.C. §103(a) as being unpatentable over Sheard et al., U.S. Patent No. 6,208,345 (Sheard), in view of Green et al., U.S. Patent No. 6,854,107 (Green).

Appellants' attorney respectfully traverses these rejections.

### B. The Appellants' Independent Claims

Independent claims 1, 9, and 17 are generally directed to developing multi-tier business applications. The computer-implemented system of claim 1 is representative, and comprises an Integrated Development Environment (IDE), executed by a computer, for creating and maintaining a multi-tier business application on a multiple tier computer network, wherein the IDE includes a Topological Multi-Tier Business Application Composer that is used by a developer to graphically create and maintain the multi-tier business application, the Composer includes a window and a palette, the palette contains graphical constructs representing tiers and components of the tiers that are used to create and maintain a graphical representation of the multi-tier business application in the window, and when creating the multi-tier business application, the developer decides on a number of tiers, identifies workstations and servers within each of the tiers, and defines processing performed by each tier and its components.

### C. The Sheard Reference

Sheard discloses a visual data integration system architecture and methodology. The system architecture includes a transport framework that represents a technology-independent integration mechanism that facilitates the exchange of technology-dependent data between disparate applications. A visual interface facilitates the design, deployment, and runtime monitoring of an integrated information system implementation. An integrated information system is developed visually through use of the visual interface by dragging and dropping components within a canvas area of the interface. The components are graphical representations of various telecommunications

hardware and software elements, such as information stores, processors, input/output devices and the like. Various components may be packaged together as business extension modules that provide specific business integration capabilities. Interconnections between components are graphically established using a mouse to define sources and destinations of specified data. An underlying configuration/runtime information framework operating above and in concert with the transport framework effectively transforms the graphical interconnections into logical or physical interconnections, which results in the contemporaneous generation of an integrated runtime system. Format neutral data meta-models are employed to model the input and output data requirements of disparate systems and system components so as to remove any cross-dependencies that exist between the systems and technologies implicated in a data integration project. The visual interface enables runtime control and analysis of the business information and system aspects of an integrated system implementation. Visual views onto the live deployment provide consistent management and control for system integrators, business integrators, system managers, and business managers using a single visual interface.

#### C. The Green Reference

Green describes a system and method for designing a software architecture for utilizing software components in building extensible N-tier software applications, the method comprising specifying a set of software component rules for creating software components; specifying a set of tier rules for creating tiers; and specifying a set of assembly rules further comprising association rules by which each tier may be associated with at least one software component and linkage rules by which each tier may be linked to at least one other tier. The tier rules may further comprise a set of association rules by which each tier created with the set of tier rules may be associated with at least one software component created using the software component rules; a set of tier framework rules to provide an architected context for software components within a tier; and a set of package rules to provide for logical grouping of interfaces within a framework defined by the tier framework rules to provide a set of specific behaviors for the tier.

- D. Arguments directed to the first grounds for rejection: Whether claims 1-24 are rendered obvious under 35 U.S.C. §103(a) by Sheard et al., U.S. Patent No. 6,208,345 (Sheard), in view of Green et al., U.S. Patent No. 6,854,107 (Green).

The Appellants' invention, as recited in independent claims 1, 9, and 17 is patentable over the references, because it contains limitations not taught by the references.

The Office Action asserts that Sheard and Green together disclose these elements at the following locations: Sheard: Col. 3, lines 12-13, Col. 49, line 1 - col. 50, line 17, Col. 3, lines 16-18, Col. 3, lines 24-26, Col. 6, lines 11-13; and Green: Col. 1, lines 16-21, Col. 4, lines 56-62 and Col. 3, lines 14-16.

The Advisory Action further asserts that Sheard and Green together disclose these elements at the following locations: Sheard: Col. 3, lines 16-18, Col. 6, lines 11-13, Col. 23, lines 10-14, Col. 24, lines 55-67, Col. 3, lines 19-26, and Col. 29, lines 32-36 and 56-60; and Green: Col. 1, lines 16-21, Col. 4, lines 56-62 and Col. 3, lines 14-16.

Appellants' attorney respectfully submits that the identified portions of Sheard and Green, taken in combination, do not render obvious Appellants' independent claims. For example, at the indicated locations, Sheard and Green merely disclose the following:

Sheard: Col. 3, lines 12-13, 16-18 and 24-26 (actually, col. 3, lines 12-44)

The present invention is directed to a visual data integration system architecture and methodology. The system architecture includes a transport framework that represents a technology-independent integration mechanism which facilitates the exchange of technology-dependent data between disparate applications. A visual interface facilitates the design, deployment, and runtime monitoring of an integrated information system implementation.

An integrated information system is developed visually through use of the visual interface by dragging and dropping component icons within a canvas area of the interface. The component icons are graphical representations of various data processing and telecommunications hardware and software elements. Various component icons may be packaged together in business extension modules to provide users with specific business integration capabilities.

Interconnections between components placed in the canvas area are graphically established using a mouse so as to define sources and destinations of specified data. An underlying configuration and runtime information framework effectively transforms the graphical interconnections into logical or physical interconnections, which results in the contemporaneous deployment of an analogous integrated runtime system. Format neutral data meta-models are employed to model the input and output data requirements of disparate systems and system components



so as to remove any cross-dependencies that exist between the systems and technologies implicated in a data integration project. The use of data meta-models in this manner effectively componentizes the systems of the data integration project, thereby permitting interconnections between system components to be established and modified using visual drag-and-drop and meta-model mapping metaphores.

Sheard: Col. 49, line 1 - col. 50, line 17

45. A computer readable medium tangibly embodying a program executable for visually implementing a data communications interface through which data passes, the data comprising informational content and a protocol, the medium comprising:

- visually depicting a first data exchange component and a second data exchange component with which only the informational content of the data is transported therebetween;

- visually linking the first data exchange component with the second data exchange component so as to visually define the data communications interface; and

- transforming the visually defined data communications interface into a runtime deployment of the data communications interface using data definition models associated with the first and second data exchange components.

46. A computer readable medium tangibly embodying a program executable for visually implementing a data communications interface through which data passes, the data comprising informational content and a protocol, the medium comprising:

- visually depicting a first data exchange component and a second data exchange component with which only the informational content of the data is transported therebetween;

- visually linking the first data exchange component with the second data exchange component so as to visually define the data communications interface; and

- verifying the utility of the visual link between the first and the second data exchange components using first and second data definition models, the first and second data definition models defining data input and data output requirements of the respective first and second data exchange components.

Sheard: Col. 6, lines 11-13 (actually, col. 6, lines 7-19)

In FIG. 1, there is illustrated a visual data integration architecture in accordance with an embodiment of the present invention. The system 30 shown in FIG. 1 provides a transport framework 33 and a visual interface 31 to facilitate the design, deployment, and runtime monitoring of an integrated information system comprising a number of disparate applications. In broad and general terms, the transport framework 33 provides a technology-independent integration mechanism that facilitates the exchange of technology-dependent data between disparate applications. The transport framework 33 enables reliable and scalable routing of information between dissimilar applications and technologies.

Sheard: Col. 23, lines 10-14 (actually lines 10-20)

In typical use, the user designs a data integration layout when the System Integration view is active by selecting various adapters and components displayed in the palette 530 of the visual interface 501. This is achieved by dragging selected adapters from the palette 530 and dropping them onto the canvas 540 using a mouse or other input device. This operation results in the creation of a new entry for the selected adapter in the project file and, additionally, results in the creation of an instance configuration file in the projects directory using a copy of the default configuration derived from the component configuration file.

Sheard: Col. 24, lines 55-67 (actually col. 24, line 51-col. 25, line 7)

The integration of data across multiple platforms and multiple workstations is coordinated through the use of a distribution planning facility. Activating the Xchange button 544 results in the presentation of a menu item which permits the user to invoke a distribution planning panel. The distribution planning panel 550, an embodiment of which is shown in FIG. 19, includes a panel 552 that provides a tree view of the network environment currently in operation for a selected data integration project. Each node in the first level 554 of the tree represents the name of a project. The second level nodes 556 under the project nodes 554 indicate the names of the workstations on which specified components are operating. A third level of nodes 558 indicates the various components operating on a particular workstation. A fourth level of nodes 560 indicates details of either component or queue elements defined on the third level of nodes 558. For example, the components shown in panel 552 of FIG. 19 includes six individual adapters, namely, CGI, ODBC, Mail, Printer, Pager, and Monitor adapters. The Monitor adapter represents a monitoring process node that is typically distinguished from other adapter nodes in terms of color or font. It is noted that the network file system mapping used to access remote machines is typically set by a system administrator outside of the visual interface environment.

Sheard: Col. 29, lines 32-36 and 56-60 (actually, col. 29, line 32 – col. 30, line 26)

As was discussed previously, a meta-model approach is used to provide a system wide specification of object and contained attribute definitions that can be used to illustrate object layout, instantiate objects, and provide for translation from one meta-defined class to another. Each adapter accepts data in a specific defined meta-model definition, manipulates the data, and produces output data in a new meta-model definition. By comparing the input and output meta-models of two interconnected adapters, it is possible to determine whether the data exchanged between the adapters is valid. Minor inconsistencies in the data requirements of two communicating adapters may be adjusted by defining mappings between the two data meta-models. Severe incompatibilities between meta-model definitions are indicative of more fundamental data issues that may require some degree of redesign to correct. The use of a meta-model approach allows the validity of a data integration

implementation to be verified, errors to be highlighted, and problems to be corrected.

Storage of the meta-model is typically implemented using a file based approach which advantageously removes any dependency on a particular database technology. Each object definition is contained in a separate file in order to isolate its definition and eliminate confusion between multiple object definitions. Each meta defined class is stored in a separate file which is named using a class plus some extension convention. The contents should be displayed in as flat a structure as possible. Each attribute consists of a single line which includes its name, type, and behavioral characteristics. Each line representing an attribute may conform to the following layout:

```
NAME|DX_DATATYPE|REQUIREMENT|RANGE
(optional)|DefaultValue(optional)
```

By way of further example, a sample configuration for an object class named Customer is provided below:

```
CustomerName|DX_STRING|MANDATORY|256|
Bank|DX_STRING|MANDATORY|256|"Rich's Bank"
AccountNumber|DX_INTEGER|MANDATORY|0-9999999|
Balance|DX_REAL|OPTIONAL||0
```

The following example is provided using the object class Customer defined above:

```
CustomerName|DX_STRING|MANDATORY||
AccountList|DX_LISTOBJECT|MANDATORY||
BEGIN:
  CheckingAcct|DX_COMMONOBJECT|OPTIONAL||
  BEGIN:
    AccountNumber|DX_INTEGER|MANDATORY|0-9999999|
    Balance|DX_REAL|OPTIONAL||0
  END:
  SavingsAcct|DX_COMMONOBJECT|OPTIONAL||
  BEGIN:
    AccountNumber|DX_INTEGER|MANDATORY|0-9999999|
    Balance|DX_REAL|OPTIONAL||0
  END:
  MoneyMktAcct|DX_COMMONOBJECT|OPTIONAL||
  BEGIN:
    AccountNumber|DX_INTEGER|MANDATORY|0-9999999|
    Balance|DX_REAL|OPTIONAL||0
  END:
END:
```

#### Green: Col. 1, lines 16-21

##### 1. Field of the Invention

The present invention relates to software design of software architectures and, in particular, to the design of a software component architecture for the

development of extensible tier software component applications, including compiled, interpreted, and on-the-fly applications.

Green: Col. 3, lines 14-16

GUID Globally unique identifier, e.g. a number having a predetermined number of bits that uniquely identifies a software component

Green: Col. 4, lines 56-62

In one embodiment, the present invention provides rules to define and create a particular N-tier architecture with a specified, initial number and type of tiers 30 and with a specified interface architecture for each tier 30, where each initial tier 30 satisfies one of a major portion of system functionality, such as business logic (processing), data, and the like.

The above portions of Sheard and Green do not teach or suggest deciding on the number of tiers, identifying workstations and servers within each of the tiers, and defining processing performed by each tier and its components, in the context of an Integrated Development Environment (IDE), that includes a Topological Multi-Tier Business Application Composer, which is used by a developer to graphically create and maintain a multi-tier business application, and includes a window and a palette, the palette contains graphical constructs representing tiers and components of the tiers that are used to create and maintain a graphical representation of the multi-tier business application in the window.

Instead, the above portions of Sheard merely describe a visual data integration system for visually linking data exchange components so as to visually define a data communications interface, while the above portions of Green merely describe the design of a software component architecture for the development of extensible tier software component applications.

Thus, the combination of Sheard and Green does not render obvious Appellants' claimed invention. Moreover, the various elements of Appellants' claimed invention together provide operational advantages over the combination of Sheard and Green. In addition, Appellants' invention solves problems not recognized by the combination of Sheard and Green.

Appellants' attorney submits that independent claims 1, 9, and 17 are allowable over the references. Further, dependent claims 2-8, 10-16, and 18-24 are submitted to be allowable over the references in the same manner, because they are dependent on independent claims 1, 9, and 17, respectively, and thus contain all the limitations of the independent claims. In addition, dependent claims 2-8, 10-16, and 18-24 recite additional novel elements not shown by the references.

VIII. CONCLUSION

In light of the above arguments, Appellants' attorney respectfully submits that the cited references do not anticipate nor render obvious the claimed invention. More specifically, Appellants' claims recite novel physical features which patentably distinguish over any and all references under 35 U.S.C. §§ 102 and 103. As a result, a decision by the Board of Patent Appeals and Interferences reversing the Examiner and directing allowance of the pending claims in the subject application is respectfully solicited.

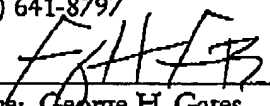
In view of the above, it is submitted that this application is now in good order for allowance and such allowance is respectfully solicited. Should the Examiner believe minor matters still remain that can be resolved in a telephone interview, the Examiner is urged to call Appellants' attorney.

Respectfully submitted,

GATES & COOPER LLP  
Attorneys for Appellants

Howard Hughes Center  
6701 Center Drive West, Suite 1050  
Los Angeles, California 90045  
(310) 641-8797

Date: August 1, 2005

By:   
Name: George H. Gates  
Reg. No.: 33,500

GHG/

## APPENDIX

1. A computer-implemented system of developing multi-tier business applications, comprising:

an Integrated Development Environment (IDE), executed by a computer, for creating and maintaining a multi-tier business application on a multiple tier computer network, wherein the IDE includes a Topological Multi-Tier Business Application Composer that is used by a developer to graphically create and maintain the multi-tier business application, the Composer includes a window and a palette, the palette contains graphical constructs representing tiers and components of the tiers that are used to create and maintain a graphical representation of the multi-tier business application in the window, and when creating the multi-tier business application, the developer decides on a number of tiers, identifies workstations and servers within each of the tiers, and defines processing performed by each tier and its components.

2. The system of claim 1, wherein the icons are dragged from the palette onto the window, and thereafter connected together, in a topological structure for the multi-tier business application.

3. The system of claim 1, wherein the components are selected from a group comprising workstations, servers, application files, connections, data paths, user-defined processes, and other user-defined elements.

4. The system of claim 1, wherein the Composer is used to perform one or more actions selected from a group comprising:

creating the tiers involved in the multi-tier business application;

specifying the components of each of the tiers; and

specifying properties that identify each of the tiers and the components of the tiers.

5. The system of claim 1, wherein the IDE further comprises a Meta-model that captures information entered via the Composer and that persistently stores the information.

6. The system of claim 5, wherein the captured information is selected from a group comprising information about tiers, workstations, servers, application files, connections, data paths, user-defined processes, and other user-defined elements.

7. The system of claim 5, wherein the Meta-model is updated and kept in synchronization with any updates made to the multi-tier business application via the Composer.

8. The system of claim 5, wherein the Meta-model is accessible by other tools.

9. A computer-implemented method for developing multi-tier business applications, comprising:

creating and maintaining a multi-tier business application on a multiple tier computer network using an Integrated Development Environment (IDE) executed by a computer, wherein the IDE includes a Topological Multi-Tier Business Application Composer that is used by a developer to graphically create and maintain the multi-tier business application, the Composer includes a window and a palette, the palette contains graphical constructs representing tiers and components of the tiers that are used to create and maintain a graphical representation of the multi-tier business application in the window, and when creating the multi-tier business application, the developer decides on a number of tiers, identifies workstations and servers within each of the tiers, and defines processing performed by each tier and its components.

10. The method of claim 9, wherein the icons are dragged from the palette onto the window, and thereafter connected together, in a topological structure for the multi-tier business application.

11. The method of claim 9, wherein the components are selected from a group comprising workstations, servers, application files, connections, data paths, user-defined processes, and other user-defined elements.

12. The method of claim 9, wherein the Composer is used to perform one or more actions selected from a group comprising:

creating the tiers involved in the multi-tier business application;

specifying the components of each of the tiers; and  
specifying properties that identify each of the tiers and the components of the tiers.

13. The method of claim 9, wherein the IDE further comprises a Meta-model that captures information entered via the Composer and that persistently stores the information.

14. The method of claim 13, wherein the captured information is selected from a group comprising information about tiers, workstations, servers, application files, connections, data paths, user-defined processes, and other user-defined elements.

15. The method of claim 13, wherein the Meta-model is updated and kept in synchronization with any updates made to the multi-tier business application via the Composer.

16. The method of claim 13, wherein the Meta-model is accessible by other tools.

17. An article of manufacture embodying logic for developing multi-tier business applications, the logic comprising:

creating and maintaining a multi-tier business application on a multiple tier computer network using an Integrated Development Environment (IDE) executed by a computer, wherein the IDE includes a Topological Multi-Tier Business Application Composer that is used by a developer to graphically create and maintain the multi-tier business application, the Composer includes a window and a palette, the palette contains graphical constructs representing tiers and components of the tiers that are used to create and maintain a graphical representation of the multi-tier business application in the window, and when creating the multi-tier business application, the developer decides on a number of tiers, identifies workstations and servers within each of the tiers, and defines processing performed by each tier and its components.

18. The article of manufacture of claim 17, wherein the icons are dragged from the palette onto the window, and thereafter connected together, in a topological structure for the multi-tier business application.



19. The article of manufacture of claim 17, wherein the components are selected from a group comprising workstations, servers, application files, connections, data paths, user-defined processes, and other user-defined elements.

20. The article of manufacture of claim 17, wherein the Composer is used to perform one or more actions selected from a group comprising:  
creating the tiers involved in the multi-tier business application;  
specifying the components of each of the tiers; and  
specifying properties that identify each of the tiers and the components of the tiers.

21. The article of manufacture of claim 17, wherein the IDE further comprises a Meta-model that captures information entered via the Composer and that persistently stores the information.

22. The article of manufacture of claim 21, wherein the captured information is selected from a group comprising information about tiers, workstations, servers, application files, connections, data paths, user-defined processes, and other user-defined elements.

23. The article of manufacture of claim 21, wherein the Meta-model is updated and kept in synchronization with any updates made to the multi-tier business application via the Composer.

24. The article of manufacture of claim 21, wherein the Meta-model is accessible by other tools.